Department of Defense

High Level Architecture

Interface Specification

Version 1.1

FINAL DRAFT

4 February 1997

TABLE OF CONTENTS

TABLE OF FIGURES

1 General

The High Level Architecture (HLA) Interface Specification is one of the three HLA definition documents. This and the other two HLA definition documents, the HLA Object Model Template and the HLA Rules, along with supporting documents, are in the HLA Technical Library on the DMSO homepage (http://www.dmso.mil).

1.1 Purpose

This document provides a specification for the DoD High Level Architecture (HLA) functional interfaces between federates and the Runtime Infrastructure (RTI). The RTI provides services to federates in a way that is analogous to how a distributed operating system provides services to applications. These interfaces are arranged into the six basic RTI service groups given below:

- Federation Management
- Declaration Management
- Object Management
- Ownership Management
- Time Management
- Data Distribution Management

The six service groups describe the interface between the federates and the RTI, and the software services provided by the RTI for use by HLA federates. The initial set of these services was carefully chosen to be those functions most likely to be required across multiple federations. As a result, federate applications will require most of the services described in this document. The RTI requires a set of services from the federate that are referred to as “RTI Initiated” and are denoted with a †.

1.2 HLA Federation Object Model Framework

A concise and rigorous description of the object model framework is essential to the specification of the interface between federates and the RTI and of the RTI services. The rules and terminology used to describe a federation object model are described in the “High Level Architecture Object Model Template” [HLA OMT]. A Simulation Object Model (SOM) describes salient characteristics of a federate to aid in its reuse and other activities focused on the details of its internal operation and as such is not the concern of the RTI and its services. A Federation Object Model (FOM), on the other hand, deals with inter-federate issues and is relevant to the use of the RTI. The DoD HLA definition states that federation object models describe:

- The set of object classes chosen to represent the real world for a planned federation,

- The set of interaction classes chosen to represent the interplay among real world objects,
- The attribute and parameters of these classes,
- The level of detail at which these classes represent the real world, including all characteristics.

Every object is an instance of an object class found in the FOM. Object classes are chosen by the object model designer to facilitate a desired organizational scheme. Each object class has a set of attributes associated with it. An *attribute* is a distinct, identifiable portion of the object state. In this discussion, “attribute designator ” refers to the attribute and “attribute value” refers to its contents. From the federation perspective, the set of all attribute values for a particular object completely defines the state of the object. Federates are free to associate additional state information with an object that is not communicated between federates, but this is outside the HLA federation object model purview.

Federates use the state of the objects as one of the primary means of communication. At any given time only one federate is responsible for simulating a given object attribute. That federate provides new values for that attribute to the other federates in the federation execution through the RTI services. The federate providing the new attribute values is said to be *updating* that attribute value. Federates receiving those values are said to be *reflecting* that attribute.

The privilege to update a value for an attribute is uniquely held by a single federate at any given time during a federation execution. A federate that has the privilege to update values for an attribute is said to *own* that attribute. The RTI provides services that allow federates to exchange ownership of object attributes. The federate that registers an object implicitly has the privilege to delete that object. The RTI provides services that allow federates to transfer the “privilegeToDeleteObject” attribute in the same way as other attributes.

All objects have an ID. The value of the ID is unique for each federation execution. Object IDs are dynamically generated by an RTI service or can be drawn from a pool of reserved values. These reserved values are set aside for special situations where federates must have knowledge of object IDs before a federation execution begins.

The FOM framework also allows for interaction classes for each object model. The types of interactions possible between different classes of objects, their affected attributes and the interaction parameters are specified. An interaction is an explicit action taken by an object, that can optionally be directed toward another object.

A *federation* is the combination of a particular FOM, a particular set of federates, and the RTI services. A federation is designed for a specific purpose using a commonly understood federation object model and a set of federates that can associate their individual semantics with that object model. A *federation execution* is an instance of executing the federation with a specific FOM, an RTI and using various execution details.

1.3 General Nomenclature and Conventions

There are various entities (classes, attributes, parameters, regions, federates, etc.) referenced in this document which can have the following different “views”:

- name - human readable
- handle - computer manipulable

The parameters to the services described in this document will use different views of the entities depending on a particular RTI implementation. For clarity, this document refers only to a generic view, known as a “designator”, when referring to these entities.

The following sets of data are needed for the implementation of a running RTI and federation executions:

- Federation Execution Data (FED) - information derived from the FOM (class, attribute, parameter names, etc.) and used by the RTI at run-time. Each federation execution needs one. In the abstract, creation of a federation execution is simply the binding of a federation execution name to a FED. The organization of FEDs will become the subject of standardization so Object Model Development Tools can automatically generate them for any vendor's RTI,
- RTI Initialization Data (RID) - RTI vendor specific information needed to run an RTI. A RID is probably supplied when an RTI is initialized.

For all federate initiated services in this specification, except 2.1, 2.2, and 2.3, there is an implied supplied parameter which is a federation execution. For all RTI initiated services there is an implied supplied parameter which is a federate. The manner in which these parameters are actually provided to the services is RTI implementation dependent, and therefore not shown in the service descriptions.

1.4 Organization Of This Document

The six HLA service groups are specified in chapters 2 through 7. Each service is described using several components:

- **Name & Description**

  service name and narrative describing the functionality of the service

- **Service Initiator**

  indicator as to whether the service is RTI or federate initiated

- **Supplied Parameters**

  required and optional service initiator provided parameters

- **Returned Parameters**

  parameters returned by the service

- **Pre-conditions**

  conditions which must exist for the service to execute correctly

- **Post-conditions**

  conditions which will exist once the service has executed correctly

- **Exceptions**

  notifications of any irregularity which may occur during service execution

- **Related Services**

  other HLA services which are related to this service

The HLA Application Programmer’s Interface (API) is given in chapter 8 in Common Object Request Broker Architecture (CORBA) Interface Definition Language (IDL) form. Chapter 9 contains the C++ API and Chapter 10 contains the Ada 95 API.

2 Federation Management

Federation Management refers to the creation, dynamic control, modification, and deletion of a federation execution.

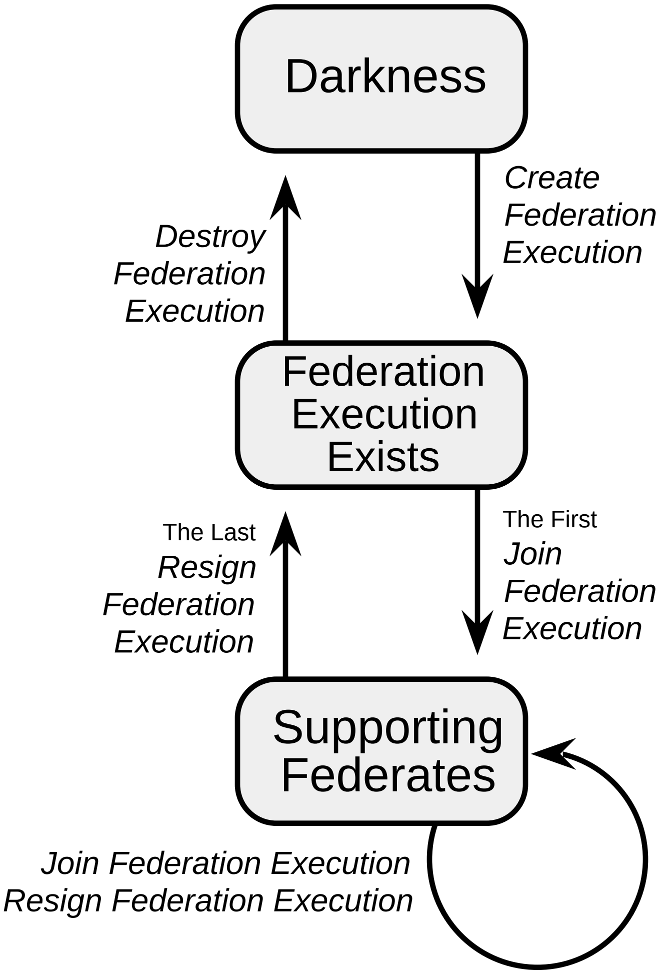
Darkness
Create Federation Execution
Destroy Federation Execution
Federation Execution Exists
The Last Resign Federation Execution
The First Join Federation Execution
Supporting Federates
Join Federation Execution
Resign Federation Execution

Figure 1. Basic States of the Federation Execution

Before a federate can join a federation execution, the federation execution must exist. Figure 1 above shows the overall state of a federation execution as certain basic federation management services are employed.

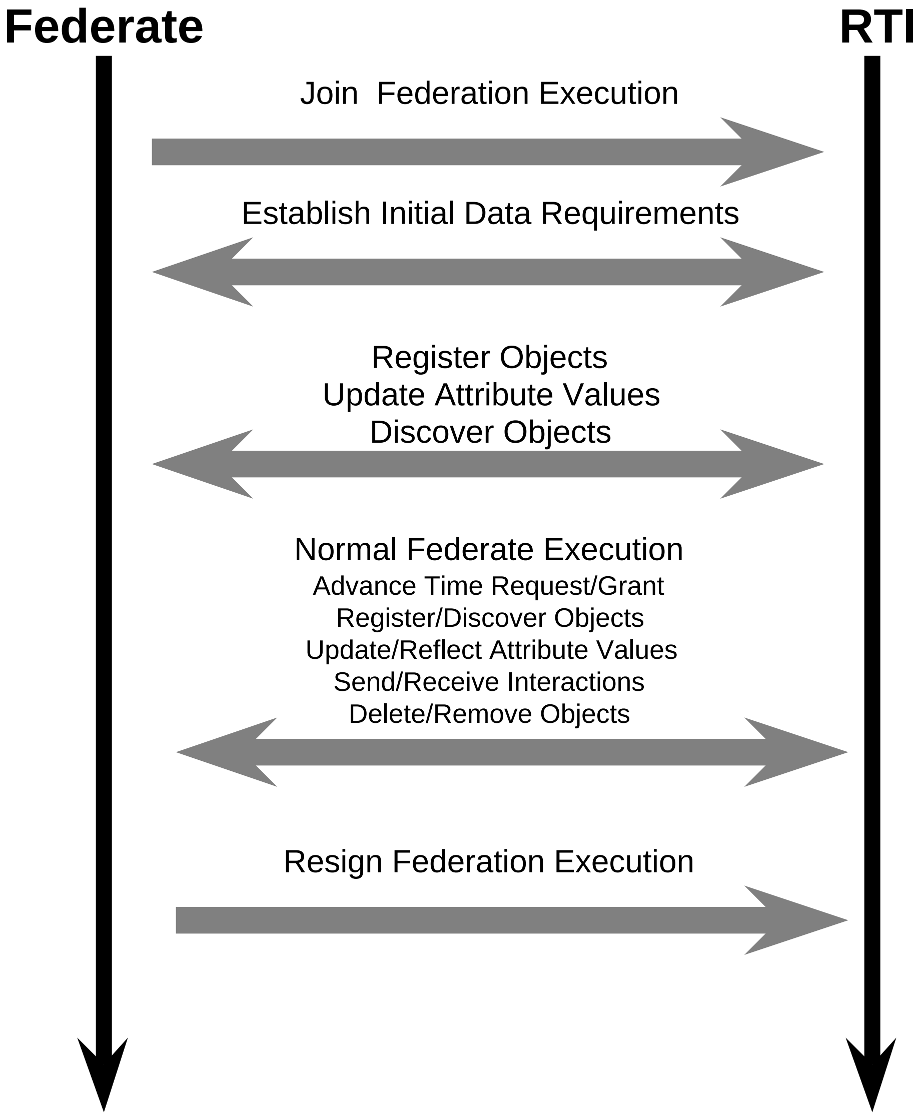
Federate
RTI
Join Federation Execution
Establish Initial Data Requirements
Register Objects
Update Attribute Values
Discover Objects
Normal Federate Execution
Advance Time Request/Grant
Register/Discover Objects
Update/Reflect Attribute Values
Send/Receive Interactions
Delete/Remove Objects
Resign Federation Execution

Figure 2. Overall View of Federate to RTI Relationship

Once a federation execution exists, federates can join and resign from it in any sequence that is meaningful to the federation user. Figure 2 above gives a generalized view of the basic relationship between a federate and the RTI during the federate participation in a federation execution. The arrows in Figure 2 represent the general invocation of RTI service groups and are not intended to demonstrate strict ordering requirements on the use of the services. The HLA concept does not preclude a single software system from participating in a given federation execution as multiple federates nor does it preclude a given system from participating in multiple (independent) federation executions.

2.1 Create Federation Execution

Federate initiated

The *Create Federation Execution* service creates a new federation execution, and adds it to the set of supported federation executions. The FED parameter supplies FOM and federation execution data to the RTI.

Supplied Parameters

A federation execution name

FED

- Set of all object class
- Set of attributes associated with each object class
- Set of interaction classes
- Set of interaction parameters associated with each interaction class
- The ordering and transportation service associated with each attribute of each object class
- The ordering and transportation service associated with each interaction class
- Optionally, routing spaces and the number of dimensions (see section 7)

Returned Parameters

None

Pre-conditions

The federation execution does not exist

Post-conditions

A federation execution exists with the given name that can be joined by federates

Exceptions

The federation execution already exists

Could not locate FED

Invalid FED

RTI internal error

Related Services

Destroy Federation Execution

2.2 Destroy Federation Execution

Federate initiated

This service removes a federation execution from the RTI set of supported federation executions. All federation activity should have stopped and all federates should have resigned before invoking this service.

Supplied Parameters

A federation execution name

Returned Parameters

None

Pre-conditions

There are no federates joined to this federation execution

Post-conditions

The federation execution does not exist

Exceptions

Federates are joined to the federation execution

The federation execution does not exist

RTI internal error

Related Services

Create Federation Execution

2.3 Join Federation Execution

Federate Initiated

The *Join Federation Execution* service affiliates the federate with a federation execution. Execution of the *Join Federation Execution* service indicates the intention to participate in the federation.

Disclaimer: The interaction between the *Join Federation Execution* service and Time Management is still under exploration. At the present time, there is not no temporal relationship between the federate and the federation at the completion of this service.

Supplied Parameters

A federate designator

A federation execution name

If required, connection parameters that allow the RTI and federate to communicate

Returned Parameters

A federate designator

Pre-conditions

The federation execution exists

The federate is not joined to that execution

Post-conditions

The federate is a member of the federation execution

Exceptions

Federate already joined to the federation execution

Specified federation execution does not exist

RTI internal error

Related Services

Resign Federation Execution

2.4 Resign Federation Execution

Federate initiated

The *Resign Federation Execution* service indicates the desired cessation of federation participation. Before resigning, ownership of attributes held by the federate should be resolved. The federate can transfer their ownership to other federates, release them for ownership acquisition at a later time, or delete the object to which they are attached. As a convenience to the federate, the *Resign Federation Execution* service accepts an action parameter that directs the RTI to perform zero, or more, of the following actions:

- delete all objects for which the federate holds that privilege
- release all other attributes for future ownership acquisition - this places the attributes into an unowned state (implying that their values are not being updated), which makes them eligible for ownership by another federate. See section 5 for a more detailed description.

Supplied Parameters

Directive to:

(1) release all attribute ownership

(2) delete all objects for which the federate holds delete privilege

(3) perform action (2) and then action (1)

(4) perform no actions

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

If optional directive is not supplied, the federate should not own any attributes

Post-conditions

The specified federate is not a member of the specified federation execution

There are no attributes in the federation execution owned by the specified federate

Exceptions

Federate owns attributes

Federate not a federation execution member

Specified federation execution does not exist

RTI internal error

Related Services

Join Federation Execution

2.5 Request Pause

Federate initiated

Indicates to the RTI the desire to stop the advance of the federation execution. The federation members will be instructed by the RTI to pause as soon after the invocation of the *Request Pause* service as possible. The label, supplied when the pause is requested, will be supplied to the other federates via the *Initiate Pause* service.

Supplied Parameters

A label

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federation execution is advancing (not paused)

Post-conditions

A federation pause is pending

Exceptions

Federation already paused

Federate not a federation execution member

RTI internal error

Related Services

Initiate Pause

Pause Achieved

Request Resume

Initiate Resume

Resume Achieved

2.6 Initiate Pause †

RTI Initiated

Instructs the federate to stop changing state as soon as possible. The label provided to the RTI when the pause was requested, via the *Request Pause* service, will be supplied to the federate.

Supplied Parameters

The label supplied when the *Request Pause* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is not already paused

Post-conditions

The federate is informed of a pending pause

Exceptions

Federate already paused

Federate internal error

Related Services

Request Pause

Pause Achieved

2.7 Pause Achieved

Federate Initiated

Indicates that the federate has successfully stopped changing state.

Supplied Parameters

The label supplied when the *Initiate Pause* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has paused

Post-conditions

The RTI is informed that the federate is paused

Exceptions

Unknown label

No pause requested

Federate not a federation execution member

RTI internal error

Related Services

Request pause

Initiate Pause

Initiate Resume

2.8 Request Resume

Federate initiated

Indicates the desire to resume the advance of the federation execution. The federation members will be instructed by the RTI to resume the advance of their state as soon after the invocation of the *Request Resume* service as possible.

Supplied Parameters

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federation execution is paused

Post-conditions

A federation resume is pending

Exceptions

Federation not paused

Federate not a federation execution member

RTI internal error

Related Services

Request Pause

Initiate Resume

Resume Achieved

2.9 Initiate Resume †

RTI Initiated

Informs a paused federate that it may return to the state evolution process in which it was engaged when it received the *Initiate Pause* service invocation. The federate should resume updating state as soon as possible.

Supplied Parameters

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is paused

Post-conditions

The federate is informed that it may resume evolving state

Exceptions

Federate not paused

Federate internal error

Related Services

Request Resume

Resume Achieved

2.10 Resume Achieved

Federate Initiated

Indicates that the federate is evolving state.

Supplied Parameters

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is evolving state

Post-conditions

The RTI has been informed that the federate is evolving state

Exceptions

No resume requested

Federate not a federation execution member

RTI internal error

Related Services

Request Resume

Initiate Resume

2.11 Request Federation Save

Federate initiated

Specifies that a federation save should take place. If the optional federation time parameter is present, the save takes place at that time. If there is no federation time parameter, the federation members will be instructed by the RTI to save as soon after the invocation of the *Request Federation Save* service as possible. The federation execution should be paused when a save occurs to help ensure consistency of the saved data among the federation participants. It is understood that the time required to perform a paused save may be unacceptable for some federations. In those cases, the federation save will be performed while the federates are executing. Only one requested/save will be outstanding at a time. A new save request replaces any outstanding save request.

Supplied Parameters

A label

Optional federation time of the desired federation save

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Save not in progress

Post-conditions

A federation save has been requested for the specified federation time

All previous requested saves are canceled

Exceptions

Save in progress

Federation time has already passed

Invalid federation time

Federate not a federation execution member

RTI internal error

Related Services

Initiate Federate Save

Federate Save Begun

Federate Save Achieved

Request Restore
Request Pause

2.12 Initiate Federate Save †

RTI Initiated

Instructs the federate to save state. If the optional federation time parameter is present, the save takes place at that time If there is no federation time parameter, the federate should save as soon after the invocation of the *Initiate Federate Save* service as possible. The label provided to the RTI when the save was requested, via the *Request Federation Save* service, will be supplied to the federate. A federate can expect a *Initiate Federate Save* invocation anytime it would expect a *Time Advance Grant* invocation. See the *Request Federation Save* service description for a discussion of the interaction of save and pause.

Supplied Parameters

The label supplied when the *Request Federation Save* service was invoked

Optional federation time to associate with the save

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

A federation save has been scheduled for time t and the RTI has determined that it is safe for the federate to advance to t

Post-conditions

The federate has been notified to begin saving its state

Exceptions

Invalid federation time

Unable to perform save

Federate internal error

Related Services

Request Federation Save

Federate Save Begun

Federate Save Achieved

2.13 Federate Save Begun

Federate Initiated

Tells the RTI that the federate is beginning to save its state. This notification allows the RTI to perform any necessary state save actions. If the federate received a federation time parameter with the corresponding *Initiate Federate Save* service invocation, that federation time should be a parameter to the *Federate Save Begun* call so the service can confirm that the correct save is being taken.

Supplied Parameters

Optional federation time associated with the save

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has received an *Initiate Federate Save* invocation

The federate is ready to start saving its state

Post-conditions

The RTI has been informed that the federate has begun saving its state

Exceptions

Save not initiated

Invalid federation time

Federate not a federation execution member

RTI internal error

Related Services

Request Federation Save

Initiate Federate Save

Federate Save Achieved

2.14 Federate Save Achieved

Federate Initiated

Tells the RTI that the federate has completed its save attempt. The save-success indicator informs the RTI that the federate save either succeeded or failed.

Supplied Parameters

A save-success indicator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has saved its state

Post-conditions

The RTI has been informed of the status of the state save attempt

Exceptions

Invalid save-success indicator

Save not initiated

Federate not a federation execution member

RTI internal error

Related Services

Request Federation Save

Initiate Federate Save

Federate Save Begun

2.15 Request Restore

Federate initiated

Directs the RTI to begin the federation execution restoration process.

Supplied Parameters

The label supplied when the *Request Federation Save* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federation has a save with the specified label

Post conditions

Federation restore is pending

Exceptions

Specified save label does not exist

Federate not a federation execution member

RTI internal error

Related Services

Initiate Restore

Restore Achieved

Request Federation Save

2.16 Initiate Restore †

RTI Initiated

Instructs the federate to return to a previously saved state indicated by the supplied federation save label.

Supplied Parameters

The label supplied when the *Request Restore* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federation has a save with the specified label

Post-conditions

The federate has been informed to begin restoring state

Exceptions

No federate save associated with the label

Could not initiate restore

Federate internal error

Related Services

Request Restore

Restore Achieved

2.17 Restore Achieved

Federate Initiated

Tells the RTI that the federate has completed its restore attempt. If restore was successful, the federate is in the state it was in when the federation save associated with the label occurred.

Supplied Parameters

The label supplied when the *Initiate Restore* service was invoked

Restore-success indicator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Federate was directed to restore through invocation of the *Initiate Restore* service

If restore was successful, the federate is in a state identical to the state it was in when the federation save associated with the supplied label occurred. If restore was unsuccessful the federate is in an undefined state.

Post-conditions

The RTI has been informed of the status of the restore attempt

Exceptions

Unknown label

Invalid restore-success indicator

Restore not requested

Federate not a federation execution member

RTI internal error

Related Services

Request Restore

Initiate Restore

3 Declaration Management

The HLA declaration management approach requires federates to declare to the RTI their desire to both generate and receive object state information. These declarations must be consistent with the Federation Object Model and made using services described in this section. In addition to object state information, the interactions generated and received by a federate must also be declared.

3.1 Publish Object Class

Federate Initiated

The information conveyed by the federate via the *Publish Object Class* service is used in multiple ways. First, it indicates which attributes of an object class the federate is capable of providing to the federation. The federate may do this by creating objects of the class and then updating the attribute values. The federate may also (or alternatively) use ownership management services to acquire attributes of objects registered by another federate (of the same object class) and then update the values of those acquired attributes. Note that every object has the "privilegeToDeleteObject" attribute that is used to manage the privilege to delete an object. This attribute is like any other attribute and its value may be updated by the owning federate if desired. A federate must publish the privilegeToDeleteObject attribute for the object class whose instances it intends to delete. If a federate intends to acquire the privilege to delete objects registered by other federates, it must both publish and subscribe to the privilegeToDeleteObject attribute for those objects' classes.

Each use of this service with the include indicator replaces all information specified to the RTI in previous "inclusive" service invocations for the same object class.

Supplied Parameters

An object class designator

Indication to include or exclude specified object class

Set of attribute designators (for include actions only)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The specified object class and attributes are part of the FED

The federate must not own object attributes of the specified object class (if the specified object class is to be excluded)

Post-conditions

The federate may now register objects of the specified class (for include only)

The federate may not register objects of the specified class (for exclude only)

Exceptions

Object class not defined in the FED

Attribute(s) not defined in the FED

Federate owns attribute(s)

Federate is not a federation execution member

RTI internal error

Related Services

Subscribe Object Class Attribute

Register Object

3.2 Publish Interaction Class

Federate Initiated

Informs the RTI which classes of interactions the federate will be sending to the federation.

Supplied Parameters

An interaction class designator

Indication to include or exclude specified interaction class

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The interaction class is specified in the FED

Post-conditions

The federate can now send interactions of the specified class (for include only)

The federate may not send interactions of the specified class (for exclude only)

Exceptions

Interaction class not defined in the FED

Federate is not a federation execution member

RTI internal error

Related Services

Subscribe Interaction Class

Send Interaction

3.3 Subscribe Object Class Attribute

Federate Initiated

Specifies (via subscribe indicator parameter) an object class for which the RTI is to begin notifying the federate of discovery of instantiated objects or specifies (via unsubscribe indicator parameter) an object class for which the RTI is to stop notifying the federate of object instance discovery.

An object instance is a member of the specified object class if it has been registered as a member of the object class or any subclass (descendent object class) of the object class as defined in the FED. Subscribing to a given object class implies a subscription to that object class and all descendent object classes. Objects registered as subclasses will be discovered by the federate as the specified object class.

When "subscribing" to an object class, the federate may also provide a set of attribute designators. The values of only the specified attributes, for all objects discovered as a result of this "subscribe" service invocation, will be provided to the federate from the RTI (via the Reflect Attribute Values service). The set of attribute designators may include attributes from the "subscribed" object class and all super-classes, as defined in the FED.

Each use of this service with the subscribe indicator replaces all information specified to the RTI in any previous "subscribe" service invocations for the same object class.

Supplied Parameters

An object class designator

Indication to subscribe to an object class (including any listed attributes) or to unsubscribe to an object class

Set of attribute designators (for subscribe action only)

An optional region designator (see section 7)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The object class and attributes are defined in the FED

The subscription region exists

Post-conditions

The RTI has been informed of the federate’s desired subscription or unsubscription

Exceptions

Object class not defined in the FED

Attribute(s) not defined in the FED

Subscription region not known

Federate is not a federation execution member

RTI internal error

Related Services

Publish Object Class

Discover Object

Reflect Attribute Values

3.4 Subscribe Interaction Class

Federate Initiated

Specifies the class of interactions which should or should not be sent to the federate.

Subscribe Interaction Class receives an interaction class designator and an indication of the federate's desire to subscribe or unsubscribe to the interaction class. Subscribing to a given interaction class implies a subscription to that interaction class and all descendent interaction classes. Each use of this service replaces all information specified to the RTI in previous service invocations for the same interaction class.

Supplied Parameters

An interaction class designator

Indication to include or exclude interaction class in subscription list

Optional subscription region designator (see section 7)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The interaction class is defined in the FED

The subscription region exists

Post-conditions

The RTI will, or will not, as indicated, deliver interactions of the specified interaction class

Exceptions

Interaction class not defined in the FED

Subscription region not known

Federate is not a federation execution member

RTI internal error

Related Services

Publish Interaction Class

Receive Interaction

3.5 Control Updates †

RTI Initiated

The RTI can issue the *Control Updates* service to the federate at any time. It tells the federate that the specified attributes for the specified object class are or are not required somewhere in the federation. If the supplied parameter indicates that the attribute values should not be updated, the federate should cease updating the values of the referenced attributes. However, if the supplied parameter indicates that the attribute values should be updated, the federate must update the values of the referenced attributes.

Supplied Parameters

An object class

A set of attribute designators

Flag indicating that the values of the specified attributes should or should not be updated

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is updating the attribute values for the object class

In the case of a request to stop updating, no federate is subscribing to the attributes of the object class

In the case of a request to update, some other federate is subscribing to the attributes of the object class

Post-conditions

The federate must or may not (as specified) update the attribute values of this object class

Exceptions

Object class not published

Attribute not published

Federate internal error

Related Services

Publish Object Class

Subscribe Object Class Attribute

Control Interactions

3.6 Control Interactions †

RTI Initiated

The RTI can issue the *Control Interactions* service to the federate at any time. It tells the federate that the specified class of interactions is or is not required somewhere in the federation. If the supplied parameter indicates that the interaction should not be sent, the federate should cease sending the referenced interaction. However, if the supplied parameter indicates that the interaction should be sent, the federate must send the referenced interaction.

Supplied Parameters

An interaction class designator

Flag indicating that the specified interactions of this class should or should not be sent

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is sending the interaction class

In the case of a request to stop sending, no federate is subscribing to the interaction class

In the case of a request to send, some federate is subscribing to the interaction class

Post-conditions

If possible, the federate will or will not (as specified) send interactions of this class

Exceptions

Interaction class not published

Federate internal error

Related Services

Publish Interaction Class

Subscribe Interaction Class

Control Updates

4 Object Management

This group of RTI services deals with the creation, modification, and deletion of objects and the interactions they produce.

4.1 Request ID

Federate Initiated

Request federation execution-unique object ID numbers. Each ID is valid for only one object registration. If the object is deleted from the federation, the ID must not be reused. The RTI will not reuse IDs once they are deleted. A set of IDs may be reserved for special purposes and will not be issued by the *Request ID* service.

Supplied Parameters

The desired number of new IDs

Returned Parameters

Set of IDs

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate can register objects with the returned IDs

Exceptions

Too many IDs requested

ID supply exhausted

Federate is not a federation execution member

RTI internal error

Related Services

Register Object

4.2 Register Object

Federate Initiated

Links an object ID with an instance of an object class. All attributes specified as publishable by the registering federate (see definition of *Publish Object Class*) are initially set as owned by the registering federate.

Supplied Parameters

An object class designator

An object ID

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is publishing the object class

The ID is a valid object ID

The ID has not already been linked with another object

Post-conditions

The specified object ID is associated with the specified object class

Exceptions

Invalid object ID

Object ID is already linked with another object

Object class not defined in FED

Federate not publishing the specified object class

Federate is not a federation execution member

RTI internal error

Related Services

Request ID

Publish Object Class

Update Attribute Values

4.3 Update Attribute Values

Federate Initiated

Provides the current attribute values to the federation for attributes owned by the federate. The federate should supply changed attribute values as specified in the FED. This service, coupled with the *Reflect Attribute Values* service, form the primary data exchange mechanism supported by the RTI.

Supplied Parameters

An object ID

A set of attribute designator and value pairs

Federation time

A user-supplied tag

Returned Parameters

An event retraction designator

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the attribute for which values are provided

Post-conditions

The RTI will distribute the new attribute values to subscribing federates

Exceptions

Object not known

Attribute(s) not defined in the FED

The federate does not own the specified attributes

Invalid federation time

Federate is not a federation execution member

RTI internal error

Related Services

Reflect Attribute Values†

Retract

Register Object

Discover Object†

Associate Update Region

4.4 Discover Object †

RTI Initiated

The *Discover Object* service informs the federate that the RTI has discovered an object. An object is discovered when the following occur:

- An attribute update is received from another federate,
- The federate has neither registered nor discovered the object,
- The update satisfies the federate’s subscription, and
- The update is in the associated region (if regions are being used, see section 7).

Supplied Parameters

An object ID

An object class designator

Federation time

A user-supplied tag

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The object class is published by some federate

The federate has subscribed to the object class

An object of that class has been registered with that ID

The federate has not discovered this object

An update attribute value has been issued by some other federate

Post-conditions

The object is known to the federate

Exceptions

The federate could not discover the object

Object class not known

Invalid federation time

Federate internal error

Related Services

Register Object

Update Attribute Values

Subscribe Object Class

4.5 Reflect Attribute Values †

RTI Initiated

Provides the federate with new values for a discovered attribute. This service, coupled with the *Update Attribute Values* service, forms the primary data exchange mechanism supported by the RTI.

Supplied Parameters

An object ID

A set of attribute designator and value pairs

Federation time

A user-supplied tag

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is reflecting the attribute values

Post-conditions

The new attribute values have been supplied to the federate

Exceptions

Object not known

Attribute not known

Invalid federation time

Federate internal error

Related Services

Update Attribute Values

Time Advance Request

Next Event Request

Time Advance Grant

4.6 Send Interaction

Federate Initiated

Informs the federation of an action taken by one object, potentially towards another object. The service returns a federation-unique event retraction designator.

Supplied Parameters

An interaction class designator

A set of parameter designator and value pairs

Federation time

A user-supplied tag

Returned Parameters

A event retraction designator

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is publishing the interaction class

Post-conditions

The RTI has received the interaction

Exceptions

Federate not publishing the specified interaction class

Interaction class not defined in FED

Interaction parameter not defined in FED

Invalid federation time

Federate is not a federation execution member

RTI internal error

Related Services

Time Advance Request

Next Event Request

Time Advance Grant

Receive Interaction

Publish Interaction Class

Retract

Associate Update Region

4.7 Receive Interaction †

RTI Initiated

Provides information about an action taken by one federation object potentially towards another object.

Supplied Parameters

An interaction class designator

Interaction parameter designator and value pairs

Federation time

A user-supplied tag

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is subscribing to the interaction class

Post-conditions

The federate has received the interaction

Exceptions

Interaction class not known

Interaction parameter not known

Invalid federation time

Federate internal error

Related Services

Retract

Send Interaction

Subscribe Interaction Class

4.8 Delete Object

Federate Initiated

Informs the federation that an object with that ID, owned by the federate, is to be removed from the federation execution. Once the object is removed from the federation execution, its ID cannot be reused. The RTI will use the *Remove Object* service to inform the reflecting federates that the object has been deleted.

Supplied Parameters

An object ID

Federation time

User-supplied tag

Returned Parameters

A event retraction designator

Pre-conditions

The federation execution exists

The federate has joined that federation execution

An object with that ID exists

The federate has the privilege to delete the object (it owns the privilegeToDeleteObject attribute)

Post-conditions

The federate no longer updates values for the object

The object does not exist in the federation execution

Exceptions

Federate does not own the delete privilege

Object not known

Invalid federation time

Federate is not a federation execution member

RTI internal error

Related Services

Remove Object

4.9 Remove Object †

RTI Initiated

Informs the federate that an object either is no longer in the associated region (if data distribution management services are used, see section 7), or the object has been deleted from the federation execution.

Supplied Parameters

An object ID

Object removal reason (deleted or out-of-scope)

Federation time

User-supplied tag

Optional event retraction designator (for deleted objects)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate knows about the object

Post-conditions

The federate has been advised to remove the object

Exceptions

Object not known

Invalid federation time

Federate internal error

Related Services

Delete Object

4.10 Change Attribute Transportation Type

Federate Initiated

The transportation type for each attribute of an object is defaulted from the object class description in the FED. A federate may choose to change the transportation type during execution. Invoking the *Change Attribute Transportation Type* service will change the transportation type for all future *Update Attribute Values* service calls for the specified attributes on the specified object.

Supplied Parameters

An object ID

A set of attribute designators

A transportation type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the attribute

Post-conditions

The transportation type is changed for the specified attributes

Exceptions

Object not known

Attribute not defined

The federate does not own the specified attributes

Invalid transportation type

Federate is not a federation execution member

RTI internal error

Related Service

Update Attribute Values

Change Attribute Order Type

4.11 Change Attribute Order Type

Federate Initiated

The data ordering type for each attribute of an object is defaulted from the object class description in the FED. A federate may choose to change the data ordering type during execution. Invoking the *Change Attribute Order Type* service will change the data ordering type for all future *Update Attribute Values* service calls for the specified attributes on the specified object.

Supplied Parameters

An object ID

A set of attribute designators

A data ordering type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the attribute

Post-conditions

The data ordering type is changed for the specified attributes

Exceptions

Object not known

Attribute not defined

The federate does not own the specified attributes

Invalid data ordering type

Federate is not a federation execution member

RTI internal error

Related Service

Update Attribute Values

Change Attribute Transportation Type

4.12 Change Interaction Transportation Type

Federate Initiated

The transportation type for each interaction is defaulted from the interaction class description in the FED. A federate may choose to change the transportation type during execution. Invoking the *Change Interaction Transportation Type* service will change the transportation type for all future *Send Interaction* service calls for the specified interaction class.

Supplied Parameters

An interaction class

A transportation type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate must be publishing the interaction class

Post-conditions

The transportation type is changed for the specified interaction class

Exceptions

Interaction class not defined

Federate not publishing the interaction class

Invalid transportation type

Federate is not a federation execution member

RTI internal error

Related Service

Send Interaction

Change Interaction Order Type

4.13 Change Interaction Order Type

Federate Initiated

The data ordering type for each interaction is defaulted from the interaction class description in the FED. A federate may choose to change the data ordering type during execution. Invoking the *Change Interaction Order Type* service will change the data ordering type for all future *Send Interaction* service calls for the specified interaction class.

Supplied Parameters

An interaction class

A data ordering type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate must be publishing the interaction class

Post-conditions

The data ordering type is changed for the specified interaction class

Exceptions

Interaction class not defined

Federate not publishing the interaction class

Invalid data ordering type

Federate is not a federation execution member

RTI internal error

Related Service

Send Interaction

Change Interaction Transportation Type

4.14 Request Attribute Value Update

Federate Initiated

The *Request Attribute Value Update* service is used to stimulate the update of values of specified attributes. When this service is used, the RTI will solicit the current values of the specified attributes from their owners using the *Provide Attribute Value Update* service. When an object class is specified, the RTI will solicit the specified attributes for all the objects of that class. When an object ID is specified, the RTI will solicit the specified attributes for the particular object. The *Request Attribute Value Update* service is intended to be used in exceptional situations, and is not intended to be used to build a "pull" system.

Supplied Parameters

Object ID or object class

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The request for the updated attribute values has been received by the RTI

Exceptions

Object not known

Object class not defined in FED

Attribute(s) not defined in the FED

Federate is not a federation execution member

RTI internal error

Related Services

Provide Attribute Value Update

Update Attribute Values

4.15 Provide Attribute Value Update†

RTI Initiated

Requests the current values for attributes owned by the federate for a given object. The federate should respond to the *Provide Attribute Value Update* service with an invocation of the *Update Attribute Values* service to provide the requested attribute values to the federation.

Supplied Parameters

An object ID

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the specified attributes

Post-conditions

The federate has been notified to provide updates of the specified attribute values

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Value Update

Update Attribute Values

4.16 Retract

Federate Initiated

Event retraction refers to the ability of a federate to retract (sometimes called cancel or unschedule) a previously scheduled event. This is a common discrete-event simulation primitive often used to model interrupts and other preemptive behaviors. Event retraction is also utilized by optimistic federates to implement mechanisms such as "anti-messages." The *Update Attribute Values* and *Send Interaction* RTI services return a designator for the event that is used to specify the event that is to be retracted. See the "HLA Time Management: Design Document" [HLA TM].

Supplied Parameters

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has previously issued *Update Attribute Values* and *Send Interaction* service calls and obtained the event retraction designators.

Post-conditions

The RTI is informed that the federate desires to retract the specified event

Exceptions

Invalid event retraction designator

Federate is not a federation execution member

RTI internal error

Related Services

Reflect Retraction

4.17 Reflect Retraction †

RTI Initiated

Event retraction refers to the ability of a federate to retract (sometimes called cancel or unschedule) a previously scheduled event. If the RTI receives a *Retract* service call for an event that has already been delivered to a federate, the *Reflect Retraction* service is invoked on the federates that received that event. See the "HLA Time Management: Design Document" [HLA TM].

Supplied Parameters

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The retracted event has been delivered to the federate

Post-conditions

The federate has been directed to retract the specified event

Exceptions

Event not known

Federate internal error

Related Services

Retract

5 Ownership Management

The ownership management group of services allows federates to transfer ownership of object attributes. An attribute is defined as a distinct and identifiable portion of the state of an object and is defined in the FED. Owning an attribute gives a federate the privilege to provide new values to the federation execution for that attribute. Additionally, ownership of the predefined attribute “privilegeToDeleteObject,” gives the owning federate the right to remove an object from the federation execution. The RTI will automatically define attribute privilegeToDeleteObject for all object instances. A federate must publish the privilegeToDeleteObject attribute for the object class whose instances it intends to delete. If a federate intends to acquire the privilege to delete objects registered by other federates, it must both publish and subscribe the privilegeToDeleteObject attribute for those objects’ classes. The value of the privilegeToDeleteObject attribute does not affect ownership management services.

In Figure 3, the federate on the left owns an object attribute that it is attempting to transfer to some other federate. In step 1, the federate invokes the *Request Attribute Ownership Divestiture* service with the object ID and designator of the attribute to be transferred and the Negotiated divestiture condition. The RTI then invokes the *Request Attribute Ownership Assumption* service on all federates that have indicated the ability to publish this attribute with the *Publish Object Class* service or, if federates were given in the *Request Attribute Ownership Divestiture* invocation, just the specified federates. Upon receipt of an *Request Attribute Ownership Assumption* invocation, a federate returns the attribute designator if it wishes to to own that attribute. The RTI selects a recipient for the attribute ownership from the set of federates that make ownership requests and completes the transfer, in step 3, with the invocation of the *Attribute Ownership Divestiture Notification* service on the divesting federate and *Attribute Ownership Acquisition Notification* on the acquiring federate. If no federates “volunteer” to own the attribute, it is up to the federate invoking the *Request Attribute Ownership Divestiture* service to “time out” and perform any appropriate actions. An “open” Negotiated *Request Attribute Ownership Divestiture* service request may be canceled by making a *Request Attribute Ownership Acquisition* call with the attributes the federate was trying to divest.

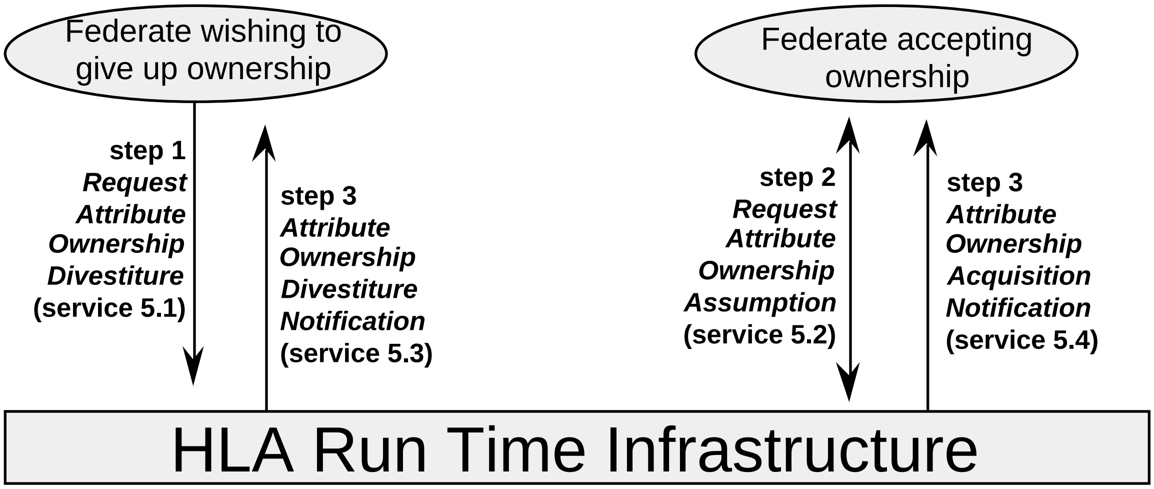
Federate wishing to give up ownership
Federate accepting ownership
step 1 *Request Attribute Ownership Divestiture* (service 5.1)
step 3 *Attribute Ownership Divestiture Notification* (service 5.3)
step 2 *Request Attribute Ownership Assumption* (service 5.2)
step 3 *Attribute Ownership Acquisition Notification* (service 5.4)
HLA Run Time Infrastructure

Figure 3. Federate Divesting Attribute Ownership (Negotiated)

If, on the other hand, a federate requests an Unconditional attribute ownership divestiture, the RTI immediately notifies the divesting federate of loss of attribute ownership via the *Attribute Ownership Divestiture Notification* service, the attribute is placed into an unowned state, and the RTI attempts to find an owner. The attribute remains in the unowned state until a) the RTI locates a federate willing to assume ownership or b) a federate requests ownership.

In Figure 4, the federate on the left is attempting to acquire ownership of an object attribute owned by the federate on the right. The federate informs the RTI of this desire, in step 1, using the *Request Attribute Ownership Acquisition* service. The RTI locates the current owner of the desired attribute, in step 2, and invokes the *Request Attribute Ownership Release* service providing the attribute owner with the designator of the desired attribute and the acquirer’s user supplied tag (perhaps indicating why the transfer is desired). If the attribute owner decides to release ownership, it returns the designator of the attribute it is releasing. Lastly, the RTI informs the acquiring federate that it now owns the attribute, step 3.

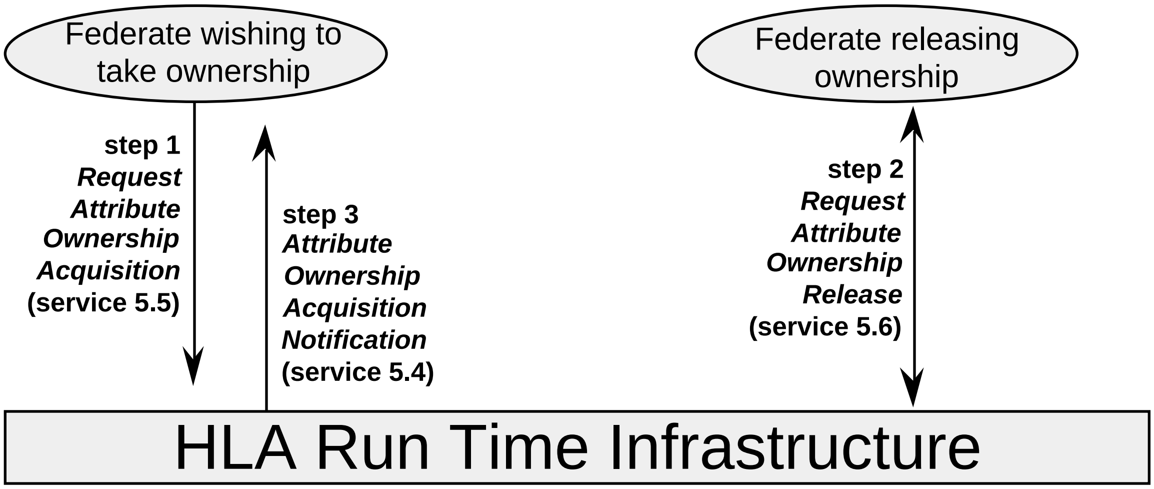
Federate wishing to take ownership
Federate releasing ownership
step 1 *Request Attribute Ownership Acquisition* (service 5.5)
step 3 *Attribute Ownership Acquisition Notification* (service 5.4)
step 2 *Request Attribute Ownership Release* (service 5.6)
HLA Run Time Infrastructure

Figure 4. Federate Acquiring Attribute Ownership

5.1 Request Attribute Ownership Divestiture

Federate Initiated

Tells the RTI that the federate no longer wants to own the specified attributes of the specified object. The federate supplies an object ID and set of attribute designators.

Options:

1. The federate can specify which federate(s) can take ownership of the released attributes, otherwise any federate may own them.
2. The federate can indicate if the desired ownership divestiture is to be negotiated or unconditional. If the divestiture is negotiated, ownership will be transferred only if some federate(s) accepts. An unconditional transfer will relieve the divesting federate of the ownership, causing the attribute(s) to go into (possibly temporarily) the unowned state, without regard to the existence of an accepting federate.

The federate must continue its publication responsibility for the specified attributes until it receives permission to stop through a positive *Attribute Ownership Divestiture Notification* service. The use of this service is illustrated in figure 3.

Supplied Parameters

An object ID

A set of attribute designators

Ownership divestiture condition (negotiated or unconditional)

A user-supplied tag

Optional set of federates

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate owns the attributes specified in the service call

Post-conditions

No change in attribute ownership

The federate has informed the RTI of its intent to divest ownership of the specified attributes

Exceptions

Object not known

Attribute(s) not defined in the FED

Federate does not own the specified attributes

Invalid divestiture condition

Invalid candidate federate

Federate is not a federation execution member

RTI internal error

Related Services

Request Attribute Ownership Assumption

Attribute Ownership Divestiture Notification

Attribute Ownership Acquisition Notification

5.2 Request Attribute Ownership Assumption †

RTI Initiated

Informs the federate that the specified attributes for the specified object are available for transfer of ownership to the federate as a result of some other federation member invoking a *Request Attribute Ownership Divestiture* service. The RTI supplies an object ID and set of attribute designators. The federate returns the subset of the supplied attribute designators for which it is willing to assume ownership. The federate will be notified as to whether or not it received ownership of the attributes in a subsequent invocation of the *Attribute Ownership Acquisition Notification* service. The use of this service is illustrated in figure 3.

Supplied Parameters

An object ID

A set of attribute designators

A user-supplied tag

Returned Parameters

Set of attribute designators (possibly empty)

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate is publishing and subscribing to the specified attributes

Post-conditions

No change in attribute ownership

The RTI has been informed of the set of attributes for which the federate is willing to assume ownership

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Attribute Ownership Divestiture Notification

Attribute Ownership Acquisition Notification

5.3 Attribute Ownership Divestiture Notification †

RTI Initiated

Notifies the federate of the result of an invocation of the *Request Attribute Ownership Divestiture* service. The service provides the federate with the set of attributes that it no longer owns. Upon this notification, the federate should stop updating the specified attribute values. The federate may receive multiple notifications for a single invocation of the *Request Attribute Ownership Divestiture* service since different federates may wish to become the owner of different attributes. The use of this service is illustrated in figure 3.

Supplied Parameters

An object ID

The set of released attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has previously attempted to divest ownership of the specified attributes

Post-conditions

The federate does not own the specified attributes

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Request Attribute Ownership Assumption

Attribute Ownership Acquisition Notification

5.4 Attribute Ownership Acquisition Notification †

RTI Initiated

Notifies the federate of the result of an invocation of the *Request Attribute Ownership Acquisition* service or *Request Attribute Ownership Assumption* service. The service supplies the set of attributes that the federate has acquired. The federate may then begin updating those attribute values. The federate may receive multiple notifications for a single invocation of the *Request Attribute Ownership Acquisition* service since various federates may wish to become the owner of different attributes. The use of this service is illustrated in figures 3 and 4.

Supplied Parameters

An object ID

A set of acquired attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate has previously attempted to acquire ownership of the specified attributes

Some other federate is willing to divest itself of ownership of the specified attributes

Post-conditions

The federate owns the specified attributes

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Request Attribute Ownership Assumption

Attribute Ownership Divestiture Notification

5.5 Request Attribute Ownership Acquisition

Federate Initiated

Requests the privilege to own the specified attributes of the specified object. The federate supplies an object ID and set of attribute designators. Before attempting to acquire ownership of an attribute, the federate should be reflecting the attribute value so it has the current attribute values. The federate may receive one or more *Attribute Ownership Acquisition Notification* invocations for each invocation of this service. The use of this service is illustrated in figure 4.

Supplied Parameters

An object ID

A set of attribute designators

A user-supplied tag

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate must be both publishing and subscribing to the specified attributes

The federate must not own the specified attributes

Post-conditions

The RTI has been informed of the federates desire to acquire ownership of the specified attributes

Exceptions

Object not known

Federate not publishing the object class

Federate not subscribing to the object class

Attribute(s) not defined in the FED

Federate not publishing the object attribute

Federate not subscribing to the object attribute

Federate already owns specified attributes

Federate is not a federation execution member

RTI internal error

Related Services

Request Attribute Ownership Release
Attribute Ownership Acquisition Notification

5.6 Request Attribute Ownership Release †

RTI Initiated

Requests that the federate release ownership of the specified attributes of the specified object. The *Request Attribute Ownership Release* service provides an object ID and set of attribute designators and is only invoked as the result of a *Request Attribute Ownership Acquisition* service invocation by some other federate. The federate returns the subset of the supplied attributes for which it is willing to release ownership. The use of this service is illustrated in figure 4.

Supplied Parameters

An object ID

A set of requested attribute designators for release

A user-supplied tag

Returned Parameters

Set of candidate attribute designators for release

Pre-conditions

The federation execution exists

The federate has joined that federation execution

The federate must own the specified attributes

Post-conditions

The RTI has been informed of the set of attributes of which the federate is willing to release ownership

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Request Attribute Ownership Acquisition

Attribute Ownership Acquisition Notification

5.7 Query Attribute Ownership

Federate initiated

The *Query Attribute Ownership* service is used to determine the owner of the specified attribute of the specified object ID. The federate supplies an object ID and attribute designator. The RTI will provide the attribute owner information via the *Inform Attribute Ownership* service call.

Supplied Parameters

An object ID

An attribute designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The request for attribute ownership information has been received by the RTI.

Exceptions

Object not known

Attribute(s) not defined in the FED

Federate is not a federation execution member

RTI internal error

Related Services

Inform Attribute Ownership

5.8 Inform Attribute Ownership †

RTI initiated

The *Inform Attribute Ownership* service is used to provide ownership information for the specified attribute of the specified object. This service is invoked by the RTI in response to a *Query Attribute Ownership* service call by a federate. This service provides the federate designator of the attribute owner (if the attribute is owned) or an indication that the attribute is available for acquisition.

Supplied Parameters

An object ID

An attribute designator

A federate designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate has been informed of the attribute ownership

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Query Attribute Ownership

5.9 Is Attribute Owned By Federate

Federate initiated

The *Is Attribute Owned By Federate* service is used to determine if the specified attribute of the specified object ID is owned by the invoking federate. The federate supplies an object ID and attribute designator. The service returns a Boolean value indicating ownership status.

Supplied Parameters

An object ID

An attribute designator

Returned Parameters

Attribute ownership indicator

Pre-conditions

The federation execution exists

The federate has joined that federation execution

Post-conditions

The federate has the requested ownership information.

Exceptions

Object not known

Attribute not known

Federate is not a federation execution member

RTI internal error

Related Services

None